U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

DOCUMENT RETRIEVAL REQUEST FORM

Requester's Name: GWEN LIANG	Case Serial Number: 10/683,075	Art Unit/Org.: 2162

Phone: 571-272-4038	Fax:	Building: RND	Room Number: 3B11

Class/Sub-Class:

Date of Request: 4-28-05	Date Needed By: ASAP

Paste or add text of citation or bibliography: Paste Citation Only one request per form. Original copy only. □

Author/Editor:			
Journal/Book Title:			
Article Title:			
Volume Number:	Report Number:		Pages:
Issue Number:	Series Number:		Year of Publication:
Publisher:			
Remarks: 143	Please see the attached page with "←" signs 15 documents 538506		

Staff Use Only

Monthly Accession Number:

Library Action	PTO		LC		NAL		NIH		NLM		NIST		Other	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
Local Attempts														
Date	4/29													
Initials	Sa													
Results	N/A													
Examiner Called														
Page Count														
Money Spent														

		Source	Date
Remarks/Comments 1st and 2nd denotes time taken to a library O/N – Under NLM means Overnight	Ordered From:	CISTI ordered* Complete faxed	
	Comments: page 574 is blank. Sheena X2-2509		

PTO-XXXX (2-96)

USCOM-DC

Document

elect the documents you wish to save or order by clicking the box next to the document,
r click the link above the document to order directly.

save locally as: PDF document search strategy: do not include the search strategy

order

143

☑ document 1 of 1 Order Document

NSPEC - 1969 to date (INZZ)

ccession number & update
3067126, C88013841; 880000.

itle
How **sensitive** is the **physical database design?** (Results of experimental investigation).

uthor(s)
Palvia-P.

uthor affiliation
Memphis State Univ, TN, USA.

ource
AFIPS Conference Proceedings. Vol.56: 1987 National Computer Conference, Chicago, IL, USA, 15-18 June 1987, p.473-82.
Sponsors: AFIPS, ACM, IEEE, SCS, Data Process. Manage. Assoc.
Published: AFIPS Press, Reston, VA, USA, 1987, xii+827 pp
Translation of: E06.

SSN
ISBN: 0-88283-051-1.

ublication year
1987.

anguage
EN.

ublication type
CPP Conference Paper.

reatment codes
P Practical.

bstract
The structure and efficiency of a **physical database design** depends on the logical data structure, the activities to take place in the **database,** the computer system characteristics, and the **physical** characteristics of the computer system. This paper identifies specific underlying factors within the broad general categories that may potentially influence the **physical database design.** In an effort to conduct a detailed sensitivity analysis of the underlying factors, an experimental **design** is developed. Sensitivity experiments are conducted as per the experimental **design,** and, finally, the experimental results are reported. (14 refs).

escriptors
data-structures; database-management-systems.

eywords
physical database design; logical data structure; sensitivity analysis.

lassification codes
C6120 (File organisation).
C6160 (**Database** management systems (DBMS)).

How sensitive is the physical database design? Results of experimental investigation

by PRASHANT PALVIA
Memphis State University
Memphis, Tennessee

ABSTRACT

The structure and efficiency of a physical database design depends on the logical data structure, the activities to take place in the database, the computer system characteristics, and the physical characteristics of the computer system. This paper identifies specific underlying factors within the broad general categories that may potentially influence the physical database design. In an effort to conduct a detailed sensitivity analysis of the underlying factors, an experimental design is developed. Sensitivity experiments are conducted as per the experimental design, and, finally, the experimental results are reported.

INTRODUCTION

The database literature has reported several research studies on selecting an optimal physical design given a set of underlying independent factors.[1,2,3,4,5] However, reports of research and experience on the sensitivity of a physical database design to the same underlying factors are practically non-existent. Such research has significant practical value to designers, who are constantly faced with restructuring databases because of changing user and technical requirements. Such findings will help designers assess the effects of major changes in the influencing factors on physical design.

This paper reports the results of sensitivity analysis based on several controlled experiments conducted in a laboratory setting. This paper includes a discussion of the objectives for physical database design and the general categories of independent factors. One of the factors is the physical design model itself, and the abstract model used for this study is described. Also, the specific factors, the factor levels and, in some cases, methods to quantify factor values are described. The experimental design is also described. The major section of the paper presents and discusses the sensitivity results from the experiments. Finally, some conclusions are presented.

PHYSICAL DESIGN OBJECTIVES AND DESIGN DETERMINANTS

Among the objectives for designing a physical database, the over-riding criterion is to minimize the operational costs of using the database (the studies referred to earlier largely use this criterion). In this study, two operational costs are considered: the cost of storing the data and the cost of accessing the data. Access costs are estimated in this paper by the surrogate measure of the total number of pages accessed from secondary memory.

The operational costs of a physical database design are influenced by four major factors: (1) the logical data structure, (2) the activities to take place on the database, (3) the computer system characteristics, and (4) the physical design model. Each category is briefly reviewed here.

The logical data structure (LDS) is designed using a logical design model, which is provided on the basis of prior design activity. The LDS for a particular design problem contains several entities and relationships joining the entities. For example, Figure 1 is the LDS of an organization's employee database. The LDS can be directly obtained using data structure diagrams[6] or by converting from entity-relationship diagrams.[7]

The activities on the database may be either retrieval or update. This work primarily focuses on retrievals. A retrieval may require selected instances of only one entity (e.g., data about certain employees only), or may require data across several entities and their instances (e.g., data about certain departments and data about employees who work in those departments). The second type of retrieval is more complex and requires "traversing" several entities.

The physical design also depends on the computer system characteristics. In a high contention multi-user environment, each access may be considered a random access, and then the total number of pages accessed can be used as a measure of access costs. The relevant computer system characteristics are the page size, the cost per page access, the storage cost, and the direct pointer size.

Finally, such physical factors as the access paths available and the data access/navigation strategy also influence the physical database design. Foremost among physical factors is the physical design model itself. The physical design model describes the permissible alternative physical designs. Different commercial DBMSs use different physical design models for the physical representation of a database. This work uses an abstract design model, which is described next.

The Physical Design Model: Record Structuring

Whereas the LDS is represented by entities and relationships between entities, the physical design is comprised of various record types, their instances, and pointer linkages between records. Additionally, access paths (e.g., indexes) may be created to permit rapid access of records.

Record structuring should be so done so as to represent the entities as well as the relationships between entities. Record structuring strategies in a file and database environment have been proposed in the literature.[1,4,5,8,9,10,11,12] A common

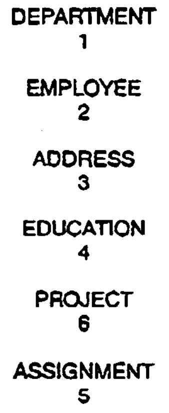


Figure 1—Example of a logical data structure

theme emerges from these works; that is, two principles are used for representing a relationship between two entities. The first principle is basic: indicate a relationship between two entities by storing appropriate pointers in the entities' instances. The pointers may be in the form of linked lists or inverted lists or some combination. The second principle for indicating a relationship is the concept of clustering/aggregation, in which all instances of one entity that are related to an instance of a second entity are clustered near the second entity instance.

The two concepts yield substantially different designs. The present abstract model captures the spirit of the two concepts; more variations and details will be included in future experiments. The abstract physical design model allows for five ways of representing a relationship between two entities X and Y: X points to Y, Y points to X, they both point to each other, X aggregates (clusters) Y, and Y aggregates X. Further, the pointers may be direct or symbolic. Aggregation of Y into X in the abstract model is actualized by making the related Y instances part of the X record.

Hierarchical and CODASYL systems incorporate the concepts of pointers and aggregations. For example, aggregation is supported in IMS by permitting hierarchical segments in the same data set, and in CODASYL systems by storing MEMBER records near the OWNER using VIA SET and NEAR OWNER. Relational systems prohibit aggregation at a logical level; however, substantial efficiencies may be achieved by its use.[13,14]

The physical options start multiplying and become more complex to evaluate as the number of entities in the LDS become large. An evaluator/simulator reported in Palvia[10] is used to evaluate the storage and access costs of any given physical database design as per the specifications of the physical design model. The evaluator is used in an exhaustive-search manner to find the optimal physical design for a given problem.

In the experiments conducted, it was soon realized that one of the three pointer options could be fairly easily selected without significantly affecting the optimal physical design. For this paper, the physical design model is simplified by permitting only one pointer option of the three options described. This option will be either mandatory two-way pointers or one-way pointers selected by the designer or by the software. With only one pointer option, a physical design can be fully specified simply by indicating the aggregations. A short-form notation devised by the author to represent a physical design is to name the "aggregator" or "absorber" (also called "parent") entity of each entity. A root entity does not have a physical parent; so its parent is numbered 0. Designs for the 6-entity problem of Figure 1 expressed in short-form include:

0 0 0 0 0 0 . . . (unclustered flat-file design)
0 1 0 2 2 0 . . . (1 clusters 2; 2 clusters 4 and 5; 1, 3 and 6 are rooted)
0 0 0 0 6 0 . . . (only 6 clusters 5; except 5, all entities rooted)

With this background, the experimental factors are described in full detail.

THE EXPERIMENTAL FACTORS

Since no assumptions are made about the sensitivity of the factors, a priori, the factors considered are comprehensive in that they make the problem character vary in most ways. The experimental factors are developed along the four exogenous dimensions; namely, the logical data structure, the activities on the database, the computer system characteristics, and the implementation characteristics. It is believed that the factors discussed in this section capture the most important features, in the spirit of the 80-20 rule, in which a relatively few number of factors tend to be the most significant.

LDS Related Factors

A. Number of entities in the logical data structure

A measure of logical data structure size and complexity is the number of entities and/or relationships in the LDS. The number of entities and the number of activities in an LDS are highly correlated, so only the number of entities is considered. Since this is the most important factor representing the LDS, four levels are chosen for this factor; that is, LDS with four, six, eight, and ten entities. The ability to generate an optimal physical design (by enumeration) prevented us from considering higher sized problems.

Activities Related Factors

The number of activities and the "structure" of activities can significantly affect the optimal physical design. Four factors relating to the structure of activities have been used. The activities related factors are: (1) the number of activities on the database, (2) the number of contexts per activity, (3) the proportion of entity instances addressed, (4) the distribution of activities on the LDS, and (5) the degree of conflict between activities.

B. Number of activities on the database

Three levels have been chosen for this factor. For the six-entity problem, the three levels are three, six, and nine activities on the database.

C. Number of contexts per activity

The number of contexts for an activity refers to the number of entities the activity traverses. The average number of contexts per activity is used as an indication of this characteristic. The three levels used are 1.67, 3.17, and 4.5 average number of contexts per activity.

D. Proportion of entity instances addressed

The proportion of entity instances addressed by an activity depends on the operating environment. For example, a production/batch environment is characterized by activities addressing a large proportion of entity instances whereas an executive environment has activities addressing only a small proportion of entity instances. Three levels have been selected, one for which 100 percent of the instances are required, one for which a medium proportion of instances are required, and one for which a very small proportion of instances are required. Call these high, medium, and low proportions.

E. Distribution of activities on the LDS

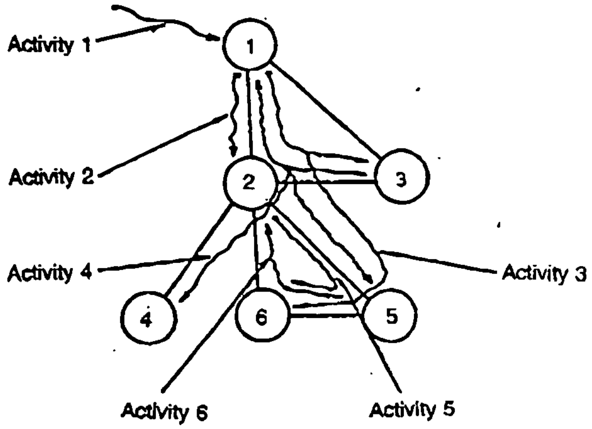
The activities on the LDS may concentrate on one or a few entities of the LDS, or may be distributed uniformly over the entire LDS. Again, use three levels: high, medium, and low concentration over the LDS. This is achieved by changing the frequencies of the various activities.

F. Degree of conflict between activities

If all activities traversed in the same direction in the LDS, there would be no conflict between activities, and it would be relatively easy to obtain the optimal physical design. In fact, it may be argued that it is the conflict among the activities which makes the design problem hard. Since a measure for the degree of conflict is not readily found in the current literature, the author developed a method to measure the degree of conflict.

In this method (see Figure 2), focus on the number of activities along each relationship. Split these activities into two categories, one for each direction along the relationship. Let $F1$ be the sum of the frequencies of all activities in one direction along a given relationship, and $F2$ be the sum of frequencies of the activities along the other direction. The greater of $F1$ and $F2$ is called FH and the smaller of the two is called FL. FH and FL are computed for each relationship in the LDS. Let FHS be the sum of all FHs and FLS be the sum of all FLs. The ratio of FLS to FHS is termed the degree of conflict. Note that this ratio varies between zero and one. A higher value represents higher conflict, while a zero value represents minimum conflict. Figure 2 also illustrates the computation of the measure of conflict.

Based on this measure, sets of activities were constructed for the six-entity problem to provide high, medium, and low degrees of conflict among activities.



Relationship	FH	FL
1-2	2	1
1-3	0	0
2-3	2	0
2-4	1	0
2-5	3	0
2-6	1	0
5-6	2	1
Total	11	2

Conflict = 2/11 = .182

Figure 2—Measure of degree of conflict

Computer System Related Factors

G. Access and storage

The relevant factor is the cost of access and storage. Three cases have been considered: only access costs, only storage costs, and a realistic case of both access and storage costs. Generally, the access costs alone are of primary importance.

H. Page size

Two levels have been used for this factor; a page size of 2000 and a page size of 4000 characters.

Implementation Factors

I. Method of accessing data

The normal mode of accessing records is to bring the records into the main memory as needed while navigating through the database. Here, this method is called AM1. A second method, called AM2, was considered for which maximum batching was assumed. AM2 requires extremely large buffer sizes so that all required records from each file are brought in all at once and the files need not be re-accessed. Unless queries are simple or main memory is very large, real systems provide a certain limit on look-foward and the extent of batching. Thus, the AM2 results should be interpreted as the asymptotic case, while the AM1 results are more realistic.

J. Random versus sequential access path

Random access paths are more suitable for today's multiuser databases requiring fast on-line retrieval speeds. Sequential access paths are also evaluated for the sake of completeness and because batch-oriented systems use sequential access paths. Thus, there are two levels.

K. Symbolic versus direct pointers

These are two levels. The mixing of pointer types is not allowed.

L. Mandatory versus non-mandatory two-way pointers

In the mandatory case, only two-way pointers are considered. In the non-mandatory case, selection among the three pointer options is made based on pair-wise consideration of entities. Thus, there are two levels.

THE EXPERIMENTAL DESIGN

For the twelve experimental factors, the total exhaustive number of cases to evaluate for a full factorial design are A(4).B(3).C(3).D(3).E(3).F(3).G(2).H(2).I(2).J(2).K(2).L (2) or 62,208 cases. For the first six factors, a new problem also has to be generated, for a total of 972 cases.

Clearly, the large number of cases precludes considering the full factorial design; the number of cases has to be reduced to a realistic proportion. Since the experiments on the evaluator are deterministic (and not stochastic), statistical methods cannot be used in developing a partial factorial design. Instead, the methodology used is to intelligently select the most important cases for experimentation. This was accomplished by first defining a "reasonable case," called the base case. The base case has each of the problem factors set at a specified value. To keep the case reasonable and average, the value of each quantitative factor is set at the middle value, and the value of each qualitative factor is set largely to reflect the current practice in database design. Table I describes the parameter values of this "average and reasonable" base case. The base case has six entities, as shown in Figure 1. (It may be noted that although six entities is not a very large number, many organizational databases can be represented by that many entities).

The next step is to change each factor, one at a time, to all of their possible values without changing the other factors. Thus, the sensitivity of each factor is tested individually without regard to each factor's interaction with other factors. If a solution is extremely sensitive to different values of a factor, then another base problem is defined by altering the value of the factor to the new value. The process is then repeated for the new base problem. In fact, Table I also lists the second base case generated in this manner. The two base cases are based on the method of accessing data, i.e., AM1 and AM2. In all, there were twenty cases for each base case, or a total of forty cases.

The philosophy of this experimental design is to evaluate the effects of each factor near the base problems. The base problems are created and recreated to assure that all significantly different experimental regions are examined.

TABLE I—Experimental factors and their levels, and parameters of the base case

Factor	Levels	Base Case Value
A. Number of entities in the LDS	4	6 entities
B. Number of activities	3	6 activities
C. Number of contexts per activity	3	Medium (3.17)
D. Proportion of entity instances addressed	3	Medium
E. Distribution of activities on LDS	3	Low concentration
F. Degree of conflict between activities	3	Medium
G. Access and storage	3	Access costs
H. Page size	2	2000 characters
I. Method of accessing data	2	AM1 for base case 1 AM2 for base case 2
J. Access Path	2	Random
K. Pointer type	2	Direct
L. Mandatory vs non-mandatory two-way pointers	2	Mandatory

SENSITIVITY ANALYSIS

Sensitivity analysis refers to the extent to which the final optimum design changes because of changes in the values of the experimental factors. The cost of the final design almost invariably changes with changes in the experimental factors. However, the change in the optimum design is the sensitivity issue and not the change in the cost of the optimum design.

The assessment of the extent of change in the optimum design remains subjective and its measurement is not readily available in the literature. To quantify this change, a measure was developed to capture the relative difference between two alternate designs. The measure determines the physical parent of each entity in the two designs and counts the number of entities with different parents. Let this number be *D*. *D* is divided by the total number of entities, *N* to obtain the physical design difference measure (PDDM). To illustrate, consider the following two physical designs:

0 0 0 2 6 0
0 1 0 2 2 0

In these two designs, entities 2 and 5 have different physical parents; thus $D = 2$. Since $N = 6$; PDDM $= 2/6 = .33$.

The optimal designs for the forty cases, found by exhaustive enumeration, are reported in Table II. Note that cases 1 to 20 are the AM1 cases and the first case is the base case for AM1; cases 21–40 are AM2 cases and the twenty-first case is the base case for AM2. As suggested earlier, the access costs are the more important costs to consider; the optimum designs minimize the access costs unless otherwise stated. In Table II, each experimental case is described by its difference from the base case. The case description is followed by the optimal design listed in its short form, followed by the operational cost of the design and the number of total designs required to be evaluated for exhaustive enumeration. To appreciate the improvement obtained by using the abstract physical design model, the commonly used operational cost of the flat-file (expressed as all zeroes in short form) design is also listed. Finally, the ratio of optimal to flat-file costs indicates the relative inefficiency of the flat-file design.

The sensitivity analysis results for the forty experimental cases are reported in Table III. Each case is compared with the base case and the PDDM value is computed. The PDDM

TABLE II—Optimal and naive design characteristics

Case	Diff. from Base Case 1 (AM1 cases)	Optimal Design			Flat-File Oper. cost	Ratio Optimal/ flat-file
		Design	Operational Cost	Designs Evaluated		
1.	None	0 1 0 2 2 0	4717	176	18869	.25
2.	4 Entities	0 1 2 2	102	20	2944	.03
3.	8 Entities	0 1 0 2 2 0 6 0	5025	956	19164	.26
4.	10 Entities	0 1 0 2 2 0 6 0 0 4	5842	5773	46656	.13
5.	3 Activities	0 0 0 0 6 0	250	176	708	.35
6.	9 Activities	0 1 0 2 2 0	5912	176	20710	.29
7.	Lo cntx/actv	0 0 0 2 0 0	443	176	623	.71
8.	Hi cntx/actv	2 3 0 2 2 0	5332	176	21499	.25
9.	Hi proportn	0 1 0 2 2 5	9658	176	53448	.18
10.	Lo proportn	0 0 0 2 6 0	543	176	858	.63
11.	Med concentr	0 1 0 2 2 0	6429	176	19337	.33
12.	Hi concentr	0 1 0 2 2 0	6634	176	19877	.33
13.	Lo conflict	0 5 0 0 6 0	9011	176	17347	.52
14.	Hi conflict	0 1 0 2 2 0	6642	176	22072	.30
15.	Storage cost	0 1 0 2 2 0	676800	176	796800	.85
16.	Acc & Strg	0 1 0 2 2 0	87.49	176	202.87	.43
17.	4000 pg sz	2 3 0 2 2 0	3407	176	18684	.18
18.	Seq acc path	2 3 0 2 2 0	38799	176	715857	.05
19.	Symbolic ptr	2 0 0 2 2 5	6739	176	18912	.36
20.	Flexible ptr	0 1 0 2 2 0	4687	176	18843	.25
	Diff. from Base Case 2 (AM2 cases)					
21.	None	0 0 0 2 6 0	593	176	642	.92
22.	4 Entities	0 1 0 2	89	20	105	.85
23.	8 Entities	0 0 0 2 6 0 6 0	682	956	790	.86
24.	10 Entities	0 0 0 2 6 0 6 0 0 4	1025	5773	1428	.72
25.	3 Activities	0 0 0 0 6 0	222	176	249	.89
26.	9 Activities	0 0 0 2 6 0	1264	176	1362	.93
27.	Lo cntx/actv	0 0 0 2 0 0	385	176	388	.99
28.	Hi cntx/actv	0 0 0 2 6 0	1238	176	1354	.91
29.	Hi proprtn	0 0 0 0 6 0	1193	176	1265	.94
30.	Lo proprtn	0 0 0 2 6 0	410	176	469	.87
31.	Med concentr	0 0 0 2 6 0	1009	176	1050	.96
32.	Hi concentr	0 0 0 0 6 0	1486	176	1515	.98
33.	Lo conflict	0 0 0 0 0 0	1090	176	1090	1.00
34.	Hi conflict	0 0 0 0 6 0	1275	176	1278	.998
35.	Storage cost	0 1 0 2 2 0	676800	176	796800	.85
36.	Acc & Strg	0 0 0 2 6 0	98.44	176	109.51	.90
37.	4000 pg sz	0 0 0 2 6 0	345	176	369	.93
38.	Seq Acc Path	0 0 0 0 6 0	1298	176	1370	.95
39.	Symbolic ptrs	0 0 0 2 0 0	696	176	723	.96
40.	Flexible ptrs	0 0 0 2 6 0	530	176	552	.96

reflects the sensitivity of a particular case. Since there are multiple cases for each factor, the combination of the multiple cases' PDDM values projects the sensitivity of the factor. Because the base case is the "middle" case of the multiple cases, the higher PDDM of the multiple cases is used as a measure of the sensitivity of the factor (this reflects the maximum change in the physical design that can occur due to change in the factor level). Note that the sensitivity rating (and PDDM) can vary between 0 and 1; where 0 means no sensitivity and 1 means maximum sensitivity. The sensitivity ratings have been classified as "high" if the rating is greater than or equal to .50, as "medium" if the rating is between .25 and .50, and as "low" otherwise.

As suggested in the experimental design section, the most sensitive factor has been the method of accessing data (i.e., AM1 vs AM2), so much so that two separate base cases were used for the two methods. For this reason, this factor's effect is explored first.

TABLE III—Sensitivity analysis results

Factor Name	Factor values (differences from base case)	AM1 Results		AM2 Results	
		PDDM	Sensitivity Rating	PDDM	Sensitivity Rating
Number of entities in the LDS	4 Entities	*		*	
	8 Entities	0	Low	0	Low
	10 Entities	0		0	
Number of activities	3 Activities	.50	High	.17	Low
	9 Activities	0		0	
Number of contexts per activity	Lo context/actv	.33	Medium	.17	Low
	Hi cntx/actv	.33		0	
Proportion of entity instances addressed	Hi proportn	.17	Medium	.17	Low
	Lo proportn	.33		0	
Distribution of activities on the LDS	Med concentr	0	Low	0	Low
	Hi concentr	0		.17	
Degree of conflict in activities	Lo conflict	.50	High	.33	Medium
	Hi conflict	0		.17	
Access and Storage	Storage cost	max	High	max	High
	Acc & Strg	of .67		of .50	
Page size	4000 pg sz	.33	Medium	0	Low
Access path	Seq acc path	.33	Medium	.17	Low
Pointer type	Symbolic ptr	.50	High	.17	Low
Mandatory 2-way vs flexible ptrs	Flexible ptrs	0	Low	0	Low

* The four-entity LDS was structurally different from the six-entity LDS, so PDDM was not calculated.

Method of Accessing Data

The method of accessing data, AM1 versus AM2, is an extremely sensitive factor. The PDDM values between corresponding cases of AM1 and AM2 (not shown here) were as high as .67. As the number of activities and the contexts per activity begin to increase, the differences between the AM1 optimal design and AM2 optimal design begin to be substantial. With few and simple activities, the AM1 and AM2 solutions are not much different. (Table II shows that the AM1 and AM2 solutions for three activities and low contexts per activity are identical.) The AM2 designs are not much sensitive to the activities on the database, but the AM1 designs are. The AM2 argument is to access each required file only once. Thus, if the file sizes are small, there will be few accesses on the file. Therefore, it may be stated that:

> *Assertion:* The objective of minimizing accesses in the AM2 (batching) case is strongly correlated to the objective of minimizing storage.

This is not so in the AM1 case where each file may be searched multiple times depending on the characteristics of the activities.

As observed earlier, a physical design technique commonly used by many designers is to store each entity independently in its own file with proper pointers to reflect relationships (e.g., in relational implementations). This design is called the flat-file design. Table II shows that, in the AM2 cases, the flat-file design is a very good design. Of the twenty AM2 cases, one flat-file design turned out to be optimal, and in fourteen cases, the cost difference between the optimal and the flat-file design was less than 10 percent. This observation is a direct corollary of the above assertion because the flat-file design is a good design from the standpoint of minimizing storage requirements (the only storage overhead is due to the pointers and no data redundancy is caused due to absorptions).

On the other hand, the flat-file design is not always good for the more commonly used access strategy as in the AM1 cases. In all of the twenty AM1 cases, the cost difference between the optimal and the flat-file design was more than 10 percent and in only one case was less than 20 percent. Further, it was found that some designs are extremely poor, costing far more than the flat-file design (on the order of 5 to 10 times more). The AM1 strategy in combination with activity factors becomes extremely sensitive and one has to be very careful in laying out the physical design. Again the reason is that in AM1, the first file is searched once, while subsequent files are searched many times. Since the flat-file calls for the maximum number of files possible, the total searches are also multiplied accordingly. This results in the flat-file being a poor design choice. The physical design selected has to minimize the total number of accesses, which is a combination of the number of searches of the files as well as the accesses at each search. This is a much more difficult design goal.

One final word on this factor. Although we have only discussed the two extremes AM1 and AM2, there are other data access strategies which fall between the two extremes (e.g., limited batching may be applied). The sensitivity of the factor is then diluted accordingly. As stated earlier, AM1 strategy is more commonly applied in most DBMSs, and the sensitivity of the underlying factors in AM1 is much higher; therefore, the remaining sensitivity results generally focus on the AM1 cases.

LDS Related Factors

The LDS related factors (i.e., the number of entities in the LDS) alter the total character of the problem. Thus, the optimal design changes to the extent the problem description changes. In a sense, it may be unfair to conduct sensitivity analysis if the changes in LDS alter the problem definition drastically. However, it does make sense to conduct sensitivity analysis by changing the LDS size without changing its basic structure (e.g., entities are merely dropped from or added to an existing LDS).

When such changes were made to the LDS of the base problem and minimal changes were made in the activities on the LDS, it was found that the optimal physical design was only moderately affected. When we experimented with adding entities to the LDS, only the physical storage of the added entities was affected with no or minimal effect on the entities previously stored. As in Table II, in both AM1 and AM2 cases, when entities 7 and 8 were added to the six-entity LDS, 7 was absorbed into 6, and 8 was stored independently. Entities 1 through 5 were stored unchanged.

LDS-based guidelines have been proposed for physical design.[1,6] The guidelines in Carlis[1] suggest that (a) a 1 : 1 relationship should be represented by pointers and (b) a 1 : M relationship should not be represented by the "M" entity absorbing the "1" entity. Evidence of the applicability of these guidelines is found in all AM2's batching cases, but not in all of the more realistic AM1 cases. For example, cases 8 and 9 violate these guidelines. Further, the experiments show that these guidelines remain valid when the activities on the database are very few and are relatively simple (e.g., each activity focusing on a very few number of entities). However, this is not the case in large multi-user databases, where there are many activities on the database and the activities may be fairly complex. It is therefore inferred that pure LDS based guidelines have limited applicability; they are applicable only when the activities on the database are few and relatively simple.

Activities Related Factors

The sensitivity experiments findings are again summarized in Table III. One of the important conclusions of the experimentation is that not only the number of activities, but also the "structure" of the activities, affect the choice of the optimal design. The amount of change in the activities related factors is a continuum, from very low to very high. The amount of change induced in the optimal design also varies similarly.

It might be said at the outset that the activity effects are more pronounced in the AM1 case than in the AM2 case. In the AM2 case, all of the activity factors had low sensitivity with one exception, and the optimal designs differed in, at most, one clustering. However, in the realistic AM1 cases, the findings were different. As expected, the sensitivity was high as the number of activities on the database increased. It is important that even when the number of activities on the database remains the same, the optimal design can change considerably with "structural" changes in the activities. These structural changes in the activities include the number of contexts per activity, the degree of conflict between entities, the proportion of entity instances addressed, and the distribution of activities on the LDS. The design changes caused by these factors are shown in Table II, and the factor sensitivity ratings are shown in Table III. As can be seen, all have a medium-to-high sensitivity to the optimal design, with the possible exception of the factor: distribution of activities on the LDS.

The sensitivity of the activity factors can be explained in an intuitive manner. Since the activities are the cause of accesses on the database, it is natural for them to be a critical factor. Clearly, when there are few activities on the database, the LDS characteristics dominate. As the number of activities increases, different design choices start to be more appealing. But, more important than the number of activities is the structure of activities. If each activity focused on one entity alone (i.e., one context activities), then a flat-file design in which each entity is placed in its own file will be a good design. However, as the contexts per activity increase, certain clusterings become desirable. For example, if an activity addresses entity B instances only via entity A instances, then it is best to cluster entity B into entity A. The proportion of entity instances addressed has the "volume" effect in that the differences due to absorption and non-absorption are multiplied according to the proportion of entity instances addressed. Finally, the effect of distribution of activities is to localize or spread the considerations of absorption/non-absorption over the LDS, thus generating different physical design choices. Perhaps the low sensitivity indicated due to this factor may be because the factor levels are not significantly apart or are not able to properly capture the factor meaning.

The last activity related factor of degree of conflict between activities makes the physical design problem especially complex. For example, consider two conflicting activities, one going from entity A to entity B and the other going from entity B to entity A. For the first activity, clustering B near A will be advantageous, while the reverse will be true for the second activity. The design choices become very sensitive as shown by the "high" sensitivity rating for the AM1 cases and the "medium" rating for the AM2 cases.

As stated earlier, designers have developed intuitive guidelines for physical design based on the logical data structure alone. In the author's opinion, this view offers a microscopic perspective on the design problem. For example, consider two LDS based guidelines suggested in Palvia.[10] The first guideline is that in a related entity pair, the entity with the higher outdegree should absorb the entity with the lower outdegree. This guideline works in most common cases, but does not work well when the activity is directed predominantly through the entity with the lower outdegree. Another guideline suggested is: if the length of an instance of the entity related to another entity is quite small in comparison, then the larger instance entity should absorb the smaller instance entity. This guideline also does not perform very well if the activity is predominantly from the smaller instance entity. Thus, as a result of these experiments, the author concludes that any physical database design guidelines or heuristics should be based both on the logical data structure and the activities to take place on the database.

Computer System Related Factors

Storage cost or access cost is a critical sensitivity factor in optimizing the physical design. This is apparent as these two are different dimensions. The storage cost for a given physical design depends only on the physical design itself; while the access cost depends both on the physical design and the activities on the database. For this reason, the "minimizing storage cost" objective function yields the same optimal solution irrespective of the other problem-related factors as long as the LDS contents and structure remain the same. On the other hand, the "minimizing access cost" objective function yields a different optimal solution based on the activity characteristics. In the AM1 cases, the PDDM values between "minimizing storage" designs and "minimizing accesses" designs ranged as high as .67.

Another factor, page size, had a medium level of sensitivity in the AM1 cases (and low in the AM2 cases). Page size has an effect on clusterings because it dictates the amount of data that can be brought into memory at once; thus, it affects the size of clusterings.

Physical Factors

The most important physical factor is the method of accessing data (discussed in a previous section). Of the remaining physical factors, mandatory versus non-mandatory two-way pointers has low sensitivity to the optimal design. Of course, the non-mandatory two-way pointers option costs less because it allows more flexibility in the direction of pointers. The low sensitivity can be easily explained because the mandatory two-way pointers automatically include the one-way pointers of the non-mandatory option.

The symbolic versus direct pointers factor had high sensitivity in the AM1 cases (and low sensitivity in the AM2 cases). As can be expected, direct pointers yield fewer page accesses because they can directly retrieve records, as opposed to going via an access path with symbolic pointers. Since direct pointers give direct address of the related record, the effect of clustering becomes largely irrelevant.

The random access path versus sequential access path factor had medium sensitivity in the AM1 cases (and again low in the AM2 cases). One would expect that absorptions would be less desirable in the sequential access paths because one would have to scan through much unnecessary data to get to the required data.

This completes the discussion of the sensitivity analysis results. As said earlier, Table III summarizes the experimental results of sensitivity analysis and may be used as a quick reference.

CONCLUSIONS

The relatively unexplored area of sensitivity of the physical database design is addressed in this paper, and contributing factors that may influence the physical database design have been identified. To study the effect of these experimental factors, a practical experimental design was developed. Based on this design, forty experimental cases, with different combinations of factor levels, were created. For each experiment, optimal physical database design was obtained using a simulation based software. Based on the experiments, the sensitivity of the optimal physical design due to changes in the factor values was analyzed.

The results of the sensitivity analysis have been reported here. An important conclusion is that activity related factors are as important in physical database design as are the logical data structure factors. The activity related factors include both the number of activities on the database as well as the structure of the activities. Several activity structural factors have been identified as sensitive factors.

Conducting sensitivity analysis of the physical database design is important, especially when restructuring the physical database. It is hoped that this exploratory study will trigger future studies as well as reports of current experience, which will validate and extend the findings of this paper.

REFERENCES

1. Carlis, J.V. "An Investigation into the Modeling and Design of Large Multi-User Databases." Ph.D. Thesis, University of Minnesota, 1980.
2. Gambino, T.J. and R.A. Gerritsen. "A Data Base Decision Support System." in *Proceedings of the VLDB,* Association for Computing Machinery, 1977.
3. Hoffer, J.A. and A. Kovacevic. "Optimal Performance of Inverted Files." *Operations Research,* 30 (1982) 2.
4. Katz, R.H. and E. Wong. "Resolving Conflicts in Global Storage Design Through Replication." *ACM Transactions on Database Systems,* 8 (1983) 1.
5. Schkolnick, M. "A Clustering Algorithm for Hierarchical Structures." *ACM Transactions on Database Systems,* 2 (1977) 1.
6. Bachman, C.W. "Data Structure Diagrams." *Data Base,* 1 (1969) 2.
7. Chen, P.P.S. "The Entity-Relationship Model—Towards a Unified View of Data." *ACM Transactions on Database Systems,* 1 (1976) 1.
8. Batory, D.S. and C.C. Gotlieb. "A Unifying Model of Physical Databases." *ACM Transactions on Database Systems,* 7 (1982) 4.
9. Maech, S.T. "Techniques for Structuring Database Records." *Computing Surveys,* 15 (1983) 1.
10. Palvia, P. "An Analytical Investigation into Record Structuring and Physical Database Design of Generalized Logical Data Structures." Ph.D. Thesis, University of Minnesota, 1984.
11. Severance, D.G. "Some Generalized Modeling Structures for Use in Design of File Organizations." *Information Systems,* 1 (1975) 2.
12. Yao, S.B. "An Attribute Based Model for Database Access Cost Analysis." *ACM Transactions on Database Systems,* 2 (1977) 1.
13. Chamberlin, D.D., et al. "History and Evaluation of System R." *Communications of the ACM,* October 1981.
14. Guttman, A. and M. Stonebraker. "Using a Relational DBMS for Computer Aided Design of Data." *IEEE Bulletin on Database Engineering,* June 1982.